sof1ane@yahoo.fr

كيفية برمجة برنامج شات بالبايثون



sof1ane@yahoo.fr

بسم الله الرحمان الرحيم

الفـــــــهرس:

1- المـــقدمة :

لابد انك في يوم احتجت لربط بين برامجك وربما استعملت ملف خارجي لعمل اتصال مثلا ملف نصي او ملف قواعد بيانات خارجي لاستقبال نص اخر من برنامج اخر

لكن هل فكرت في طريقة اخرى ؟؟

نعم توجد طريقة اخرى وهي باستخدامsocket

لكن ما معنى socket؟

هي التقنية والوسيلة الفعالة التي تمكننا من عمل شبكة بين تطبيقات سواء شبكة محلية او على الانترنت

وهي تستعمل كثيرا في الالعاب (jeux en ligner)و كل ما هو متعلق بالربط بين تطبيقات،

برمجة socketمتوفرة في الكثير من لغات البرمجة سي، جافا

...

فكما تعلم تعتبر لغة البايثون من اقوى اللغات في التعامل مع الشبكات وبرمجتها بحيث انها توفر العديد من المكتبات التي المتخصصة في ذالك من هاته المكتبات مكتبة socket سنتمكن من خلالها من كتابة برامج تستطيع الأتصال عبر الشبكة

"low socket programming" .

اريد ان اوضح في البداية برامترين مهمين في socket

<u>address_family:</u>

AF_INET: العناوين الخاصة ببرتوكول IP الإصدارة الرابعة

AF_INET6 : IP العناوين الخاصة ببرتوكول الإصدارة السادسة.

AF_UNIX . : هذه العناوين خاصة بأنظمة لينكس

protocol:

البرتوكول الذي سيتم أستخدامه في الأتصال

SOCK_STREAM : برتوكول *TCP*هذا الذي سنعمل به

UDP برتوكول:SOCK_DGRAM

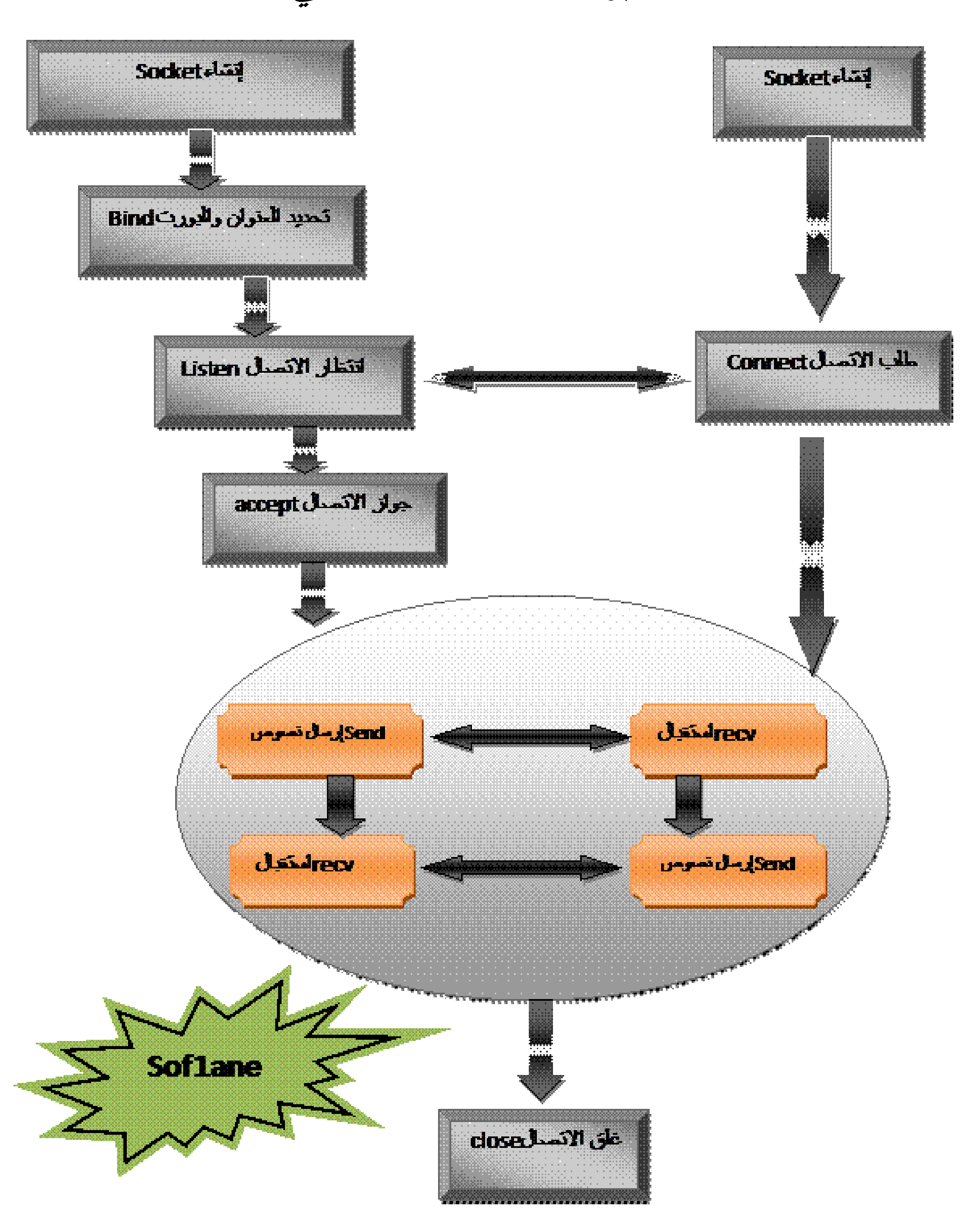
توجد فرق بينهم

لكن لا يمكن الخوض فيهم لكي لا نخرج من موضعنا

sof1ane@yahoo.fr

المهم

و مهما اختلفت اللغات و تعقدت يبقى المبدأ واحد كما سيوضحه المخطط الأتي



2- برمجة سكريبت الخادم :

ملاحظات :

-قمت باستخدام البرمجة الموجهة (كلاس ، ميتود = طريقة) لكن يمكن استخدام برمجة الإجرائية في برمجته

المهم هو فهم المبدأ.

-استعملت في السكريبت ip : 127.0.0.1 هو للجهاز المحلي يعني لجهازك الخاص و هو اي بي المرافق للعنوان التالي localhost.

Python

```
import socket //هنا نقوم باستدعاء المكتبة
class sof1ane_serveur(object): //نقوم بتعريف كلاس جديد
   def __init__(self, host='', port=8080): //هنا المشيد والبرامترات الخاصة
      self._host, self._port=host, port //ارجاع قيم الهوست و بورت
      self.address=(host, port)

                  // متغير من نوع قائمة وارجاع قيم الهوست وبورت فيه

      self.sock=socket.socket(socket.AF_INET, socket.SOCK_STREAM)


                              //ها نقوم بتحديد كل من

               socket types et address (and protocol) families

      self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

            //هنا يجب تفعيل SO_REUSEADDR ليتم غلق البورت بعد غلق السيرفر

      عدم استخدام هذا المر يؤدي الى حدوث خطأ في تشغيل السكريبت مرة اخرى تحت عنوان البورت مشغول


   def demarrer(self): // تعريف ميتود جديد
```

```
        self.sock.bind(self.address)
```

ربط العنوان الخاص بالخادم الذي يتم تنفيذ البرنامج عليه //

```
        self.sock.listen(1)
```

تجهيز الخادم لاستقبال الاتصالات منالشبكة مع تحديد عدد معين للاتصالات لاستقبالها//

```
        print "Server s'executant sur: ", self._port
        self.gerer_con()
```

نستدعي الطريقة تسيير الاتصال الذي سنتطرق اليها //

```
    def handle_request(self):
```

تعريف الطريقة تسيير الاتصال //

```
        while True:
```

حلقة غير منتهية لانتظار الاتصال//

```
            clientsock, addr=self.sock.accept()
```

لحضة قبول الأتصال نستقبل عنوان الأي بي الخاص بالعميل و مع توفير منفذ محلي//

ملاحظة للطريقة accept() اهمية كبيرة نضرا لانها ترجع لنا كائن العميل مع عنوانه

```
            print "Connexion à partir de: ", addr
            clientsock.sendall(str(addr)+" vous etes connecte au serveur...")
```

ارسال رسالة نجاح الاتصال //

```
            while True:

                msg=clientsock.recv(100)
```

تستخدم send et recv الطريقة مع الكائن الذي انشأته//

accept() و 100 تمثل عدد الاحرف القصوى في كل رسالة

```
                if msg:
                    print ">> ", msg
                clientsock.sendall(msg)
```

ارسال رمز >> //

الذي يبين جاهزية استقبال الرسائل للعميل

```
        clientsock.close()
if __name__=="__main__":
    try:
        serveur=sof1ane_serveur()
        serveur.start()
    except KeyboardInterrupt:
        exit()

تنفيذ البرنامج الرئيسي مع التعامل مع الاخطاء في حال حدوثها //
```

3-برمجة سكريبت العميل :

Python

```
import socket //هنا نقوم باستدعاء المكتبة
class SimpleClient(object)://نقوم بتعريف كلاس جديد
    def __init__(self, endpoint=('127.0.0.1', 8080)):

//هنا المشيد والبرامترات الخاصة//
        self._endpoint=endpoint
//ارجاع قيم الهوست و بورت//

        self.sock=socket.socket(socket.AF_INET, socket.SOCK_STREAM)

//ها نقوم بتحديد كل من//

socket types et address (and protocol) families

المشروحة في المقدمة في الموضوع الأول

        self.sock.connect(self._endpoint)
//connect نقوم بعمل الاتصال مع السيرفر عن طرق المنهج

    def start(self):
        while True:
// نقوم بفتح حلقة اتصال مع السيرفر

            data=self.sock.recv(8096)
//ارسال و استقبال البيانات//

            if not data:
                break
            print data
            msg=raw_input("> ")
            if not msg:
```

```
            break
          self.sock.send(msg)
// غلق الاتصال

        self.sock.close()
if __name__=="__main__":

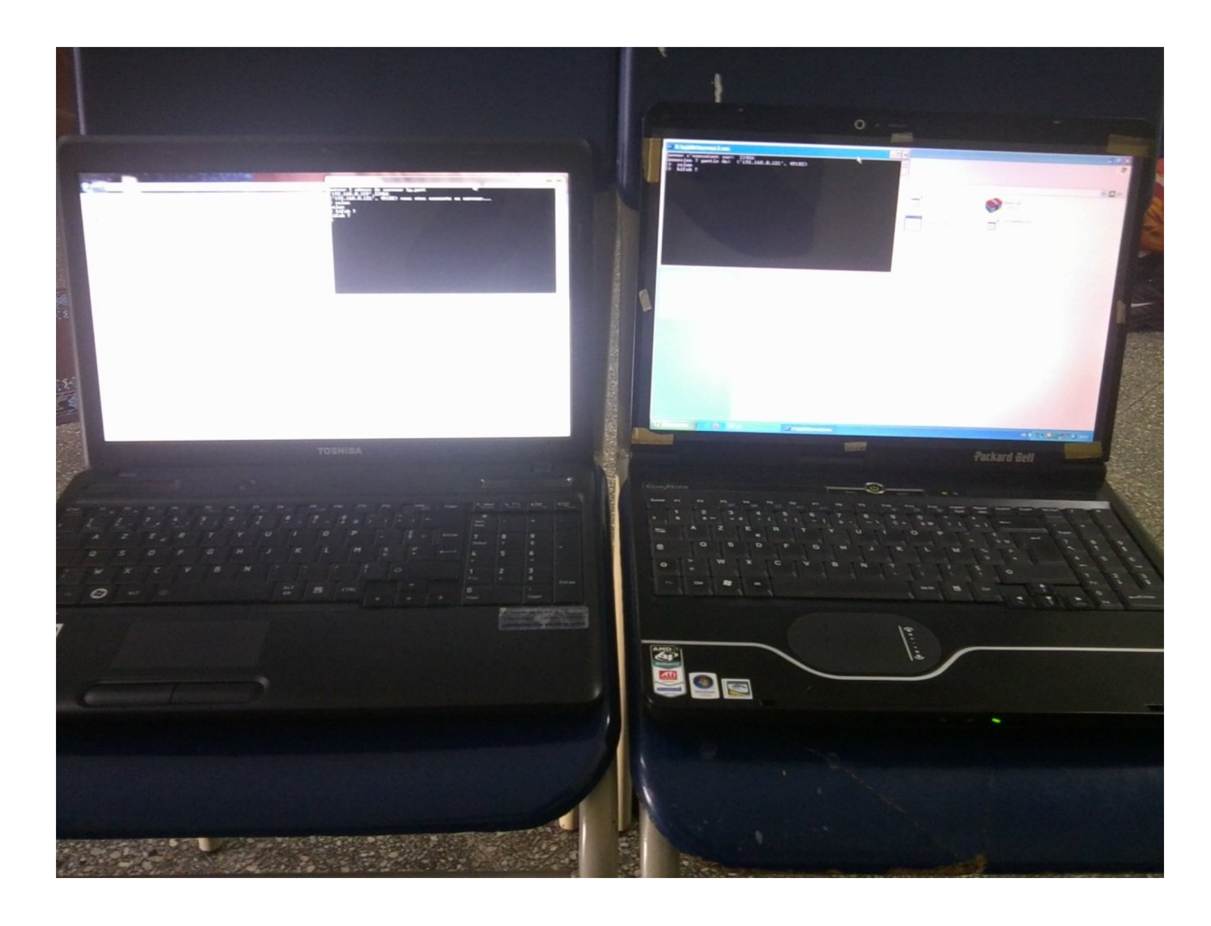
// تنفيذ البرنامج الرئيسي مع التعامل مع الاخطاء في حال حدوثها

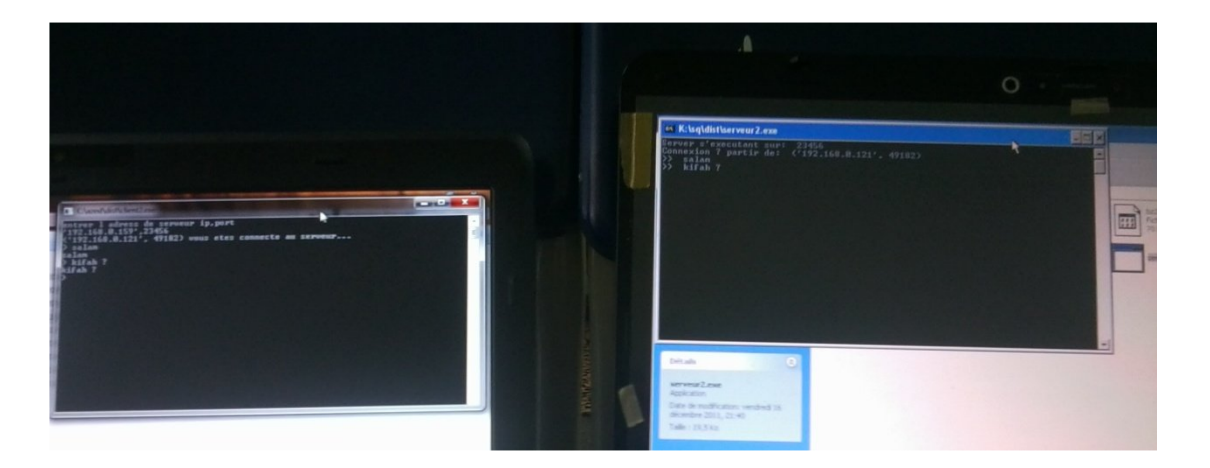
    try:
        sc=SimpleClient()
        sc.start()
    except KeyboardInterrupt:
        exit()
```

4-تطبيق البرنامج في شبكة محلية :

هنا لا نحتاج سوى تحديد عنوان اي بي والبورت

الصور





5-استعمال تقنية Threading:

5.1- برمجة سكريبت الخادم :

نلاحظ حين القيام بفتح حلقة اتصال مع السيرفر يكون البرنامج في حالة انتظار لايمكن لا ارسال له و لا عمل أي عمل اخر بحيث يقوم المعالج بفتح الحلقة اتصال وبس .

لتوضيح اكثر مثلا اذا طلبت من المستعمل كتابة اسمه عن طريق inputيبقى السكريبت متوقف الى غاية ادخال اسمه لا يمكنه عمل اي عملية اخرى و هذا يعد عيب في البرنامج .

سنحاول في تطبيقنا باستعمال تقنية threadلكي نخبر المعالج ان ينفذ كل عمليه في (process) منفصل

أي يمكن فتح أكثر من حلقة في نفس الوقت او مثلا فتح حلقة و في نفس الوقت اظهار نص مثلا .

```
import socket,threading
class sof1ane_serveur(object):
    def __init__(self, host, port):
        self._host, self._port=host, port
        self.address=(host, port)
        self.sock=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    def start(self):
        self.sock.bind(self.address)
        self.sock.listen(5)
        print "Server s'executant sur: ", self._port
        self.gerer_con()
    def gerer_con(self):
        try:
            while True:
                clientsock, addr=self.sock.accept()
                print "Connexion ? partir de: ", addr
                clientsock.sendall(str(addr)+" vous etes connecte au serveur...")
                self.th1 = threading.Thread(target=self.thread1, args=[clientsock])
                self.th2 = threading.Thread(target=self.thread2, args=[clientsock])
                self.th1.start()
                self.th2.start()
        except:
            exit()
    def thread1(self,clientsock):
        try:
            while True:
                msg=clientsock.recv(2048)
                if not msg:
                    break
                print ">", msg
            self.sock.close()
        except:
            exit()
    def thread2(self,clientsock):
        try:
            while True:
                mesg = raw_input(">")
                if not mesg:
                    break
                clientsock.send('Serveur dit :'+mesg)
            self.sock.close()
        except:
            exit()
if __name__=="__main__":
    try:
        serveur=sof1ane_serveur('',23456)
        serveur.start()
    except:
        exit()
```

5.2- برمجة سكريبت الزبون :

```
1.  import socket,threading
2.  class project(object):
3.  def __init__(self,ip_port):
            self.ip_port = ip_port
            self.sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
            self.sock.connect(self.ip_port)
4.  def recev(self):
        while 1:
        msg_rec = self.sock.recv(8096)
        if not msg_rec:
            break
        print msg_rec
5.  self.sock.close()
6.  def env(self):
        while 1:
        msg_env = raw_input("==>")
        if not msg_env:
            break
        self.sock.send(msg_env)
        self.sock.close()
7.  if __name__=="__main__":
8.  print("Saisie l'adress IP et le port")
9.  ip_port_saisie=input()
10. object_projet = project(ip_port_saisie)
11. thread1 = threading.Thread(target=object_projet.env, args=[])
12. thread2 = threading.Thread(target=object_projet.recev, args=[])
13. thread1.start()
14. thread2.start()
```

sof1ane@yahoo.fr

بعض الصور لمعالجة اكثر من اربعة اتصالات

